

LISTING OF AND AMENDMENTS TO CLAIMS:

1. - 3. (Canceled)

4. (currently amended) A method of mapping data formatted according to either of a group consisting of a relational database model and an Abstract Syntax Notation ("ASN.1") data model into an Extensible Markup Language ("XML") compliant data format, the method including an exploration/adaptation step for exploration of legacy data in a legacy database on a data resource, and a production step for generation of XML data out of the legacy data, wherein the method performs data format mapping between the legacy database and an XML-compliant representation of that data, and wherein the data format mapping takes ASN.1 constructs, and performs the following steps:

(a) mapping primitive ASN.1 types onto XML entities containing character data, wherein, for each primitive ASN.1 type, one entity is defined;

(b) redefining fields of ASN.1 constructs, which are of a primitive ASN.1 type, as XML elements with an attribute of the corresponding entity;

(c) mapping the ASN.1 ANY type onto an XML entity containing uninterpreted data;

(d) mapping ASN.1 constants onto XML entities;

(e) mapping the ASN.1 SEQUENCE construct onto an XML element containing individual elements of the SEQUENCE as an XML sequence, wherein the XML "?" operator is applied to optional ASN.1 elements;

(f) mapping the ASN.1 SEQUENCE OF construct onto the XML repetition construct of elements "\*";

(g) mapping the ASN.1 SET construct onto the XML CHOICE construct, wherein the XML "?" operator is applied to optional ASN.1 elements in a SET;

(h) mapping the ASN.1 SET OF construct onto the XML repetition construct of "\*";

(i) mapping the ASN.1 CHOICE construct onto an XML element containing individual elements of the CHOICE as XML alternatives by using the "|" operator;

(j) mapping the ASN.1 COMPONENTS OF construct as follows:

for every COMPONENTS OF construct, creating an XML element which contains individual component elements, wherein the COMPONENTS OF optionally has an attribute that refers to the contained construct; and

(k) mapping each ASN.1 module having certain contents onto a specific XML Document Type Description by recursively applying the above rules to the module's contents, wherein each Document Type Description forms a separate name space and the ASN.1 constructs IMPORT and EXPORT are then modeled using qualified names for imported/exported XML elements, thus creating an XML specification.

2  
5. (original) The method of claim 4, wherein in the exploration step, standardized database functions are used to retrieve information on a database's data scheme.

3  
6. (original) The method of claim 4 wherein the data format mapping is automated between legacy databases and an XML-compliant representation of that data.

4  
7. (currently amended) The method of claim <sup>3</sup> 6, wherein the data format mapping submethod covers all ASN.1 constructs, both primitive and composite.

5  
8. (currently amended) The method of claim <sup>1</sup> 4 [[2]] wherein an XML Document Type Definition (~~"DTD"~~) conversion between either of the group and an XML-compliant format is made.

9. (currently amended) A method that automates the data format mapping between data in a legacy database on a data resource formatted according to one member of a group selected from relational databases modeled according to a relational data model and Extensible Markup Language ("XML")-compliant representations of that data, wherein the method comprises:

(1) an exploration/adaptation step for exploration of the data resource, and

(2) a mapping step for generation of XML data out of the legacy data, wherein the data format mapping takes ASN.1 constructs, and performs the following steps:

(a) mapping primitive ASN.1 types onto XML entities containing character data, wherein, for each primitive ASN.1 type, one entity is defined;

(b) redefining fields of ASN.1 constructs, which are of a primitive ASN.1 type, as XML elements with an attribute of the corresponding entity;

(c) mapping the ASN.1 ANY type onto an XML entity containing uninterpreted data;

(d) mapping ASN.1 constants onto XML entities;

Sub  
B/S

(e) mapping the ASN.1 SEQUENCE construct onto an XML element containing individual elements of the SEQUENCE as an XML sequence, wherein the XML "?" operator is applied to optional ASN.1 elements;

(f) mapping the ASN.1 SEQUENCE OF construct onto the XML repetition construct of elements "\*";

(g) mapping the ASN.1 SET construct onto the XML CHOICE construct, wherein the XML "?" operator is applied to optional ASN.1 elements in a SET;

(h) mapping the ASN.1 SET OF construct onto the XML repetition construct of "\*";

(i) mapping the ASN.1 CHOICE construct onto an XML element containing individual elements of the CHOICE as XML alternatives by using the "|" operator;

(j) mapping the ASN.1 COMPONENTS OF construct as follows:

AI

for every COMPONENTS OF construct, creating an XML element which contains individual component elements, wherein the COMPONENTS OF optionally has an attribute that refers to the contained construct; and

(k) mapping each ASN.1 module having certain contents onto a specific XML Document Type Description by recursively applying the above rules to the module's contents, wherein each Document Type Description forms a separate name space and the ASN.1 constructs IMPORT and EXPORT are then modeled using qualified names for imported/exported XML elements, thus creating an XML specification.

Sub  
B7  
10. (currently amended) The method of claim 9 wherein the mapping step includes the following substeps:

(a) mapping relations, ~~such as tables~~, within the relational data model onto XML elements including a "table" and a "row" element;

(b) mapping tuples, ~~such as rows~~, within a relation onto XML elements which are nested within the XML "table" element; and

(c) mapping attributes ~~(i.e., columns)~~ of tuples onto XML elements which are nested within the XML "row" element.

11. Canceled

AI  
11  
12. (currently amended) The method of claim 9 [[11]], wherein in order to perform a lossless mapping onto XML, further inserting ASN.1 tags into the XML specification by including one TAG element and three TAG attributes in the XML specification.

12  
11  
13. (original) The method of claim 12, wherein the XML TAG elements use an attribute to indicate their policy, their nature, and value of the tag and wherein, an additional attribute indicates whether IMPLICIT or EXPLICIT ASN.1 tagging has been used.

13  
12  
14. (currently amended) The method of claim 13 wherein additional attributes are included in the XML specification, wherein XML tag attributes in a separate name-space Abstract Syntax Notation are ASN:policy ~~(EXPLICIT or IMPLICIT)~~, ASN:class ~~(UNIVERSAL, APPLICATION,~~

CONTEXT, ~~PRIVATE~~}, and ASN:tag containing the original ASN.1 tag value.

[15. - 18. (Canceled)

14

19. (currently amended) A computer-readable medium encoded with a method of mapping data formatted according to either of a group consisting of a relational database model and an Abstract Syntax Notation ("ASN.1") data model into an Extensible Markup Language ("XML") compliant data format, the method including an exploration/adaptation step for exploration of legacy data in a legacy database on a data resource, and a production step for generation of XML data out of the legacy data, wherein the method performs data format mapping between the legacy database and an XML-compliant representation of that data, and wherein the format mapping takes ASN.1 constructs, and performs the following steps:

(a) mapping primitive ASN.1 types onto XML entities containing character data, wherein, for each primitive ASN.1 type, one entity is defined;

(b) redefining fields of ASN.1 constructs, which are of a primitive ASN.1 type, as XML elements with an attribute of the corresponding entity;

(c) mapping the ASN.1 ANY type onto an XML entity containing uninterpreted data;

(d) mapping ASN.1 constants onto XML entities;

(e) mapping the ASN.1 SEQUENCE construct onto an XML element containing individual elements of the SEQUENCE as an XML sequence, wherein the XML "?" operator is applied to optional ASN.1 elements;

(f) mapping the ASN.1 SEQUENCE OF construct onto the XML repetition construct of elements "\*";

(g) mapping the ASN.1 SET construct onto the XML CHOICE construct, wherein the XML "?" operator is applied to optional ASN.1 elements in a SET;

(h) mapping the ASN.1 SET OF construct onto the XML repetition construct of "\*";

(i) mapping the ASN.1 CHOICE construct onto an XML element containing individual elements of the CHOICE as XML alternatives by using the "|" operator;

(j) mapping the ASN.1 COMPONENTS OF construct as follows:

for every COMPONENTS OF construct, creating an XML element which contains individual component elements, wherein the COMPONENTS OF optionally has an attribute that refers to the contained construct;  
and

(k) mapping each ASN.1 module having certain contents onto a specific XML Document Type Description by recursively applying the above rules to the module's contents, wherein each Document Type Description forms a separate name space and the ASN.1 constructs IMPORT and EXPORT are then modeled using qualified names for imported/exported XML elements, thus creating an XML specification.

15  
20. (original) The medium of claim 19, wherein in the exploration step, standardized database functions are used to retrieve information on a database's data scheme.

16  
21. (original) The medium of claim 14 wherein the data format mapping is automated between legacy databases and an XML-compliant representation of that data.

17  
22. (currently amended) The medium of claim 21, wherein the data format mapping submethod covers all ASN.1 constructs, both primitive and composite.

21  
23. (currently amended) The medium of claim 17 wherein an XML Document Type Definition (~~"DTD"~~) conversion between either of the group and an XML-compliant format is made.

22  
24. (currently amended) A medium that automates the data format mapping between data in a legacy database on a data resource formatted according to one member of a group selected from relational databases modeled according to a relational data model and Extensible Markup Language ("XML")-compliant representations of that data, wherein the method comprises:

[[a]] an exploration/adaptation step for exploration of the data resource, and

[[b]] a mapping step for generation of XML data out of the legacy data, and wherein the format mapping takes ASN.1 constructs, and performs the following steps:

(a) mapping primitive ASN.1 types onto XML entities containing character data, wherein, for each primitive ASN.1 type, one entity is defined;

(b) redefining fields of ASN.1 constructs, which are of a primitive ASN.1 type, as XML elements with an attribute of the corresponding entity;



(c) mapping the ASN.1 ANY type onto an XML entity containing uninterpreted data;

(d) mapping ASN.1 constants onto XML entities;

(e) mapping the ASN.1 SEQUENCE construct onto an XML element containing individual elements of the SEQUENCE as an XML sequence, wherein the XML "?" operator is applied to optional ASN.1 elements;

(f) mapping the ASN.1 SEQUENCE OF construct onto the XML repetition construct of elements "\*";

(g) mapping the ASN.1 SET construct onto the XML CHOICE construct, wherein the XML "?" operator is applied to optional ASN.1 elements in a SET;

(h) mapping the ASN.1 SET OF construct onto the XML repetition construct of "\*";

(i) mapping the ASN.1 CHOICE construct onto an XML element containing individual elements of the CHOICE as XML alternatives by using the "|" operator;

(j) mapping the ASN.1 COMPONENTS OF construct as follows:

for every COMPONENTS OF construct, creating an XML element which contains individual component elements, wherein the COMPONENTS OF optionally has an attribute that refers to the contained construct;  
and

(k) mapping each ASN.1 module having certain contents onto a specific XML Document Type Description by recursively applying the above rules to the module's contents, wherein each Document Type Description forms a separate name space and the ASN.1 constructs IMPORT and EXPORT are then modeled using qualified names for

imported/exported XML elements, thus creating an XML specification.

<sup>23</sup>  
25. (currently amended) The medium of claim <sup>22</sup>24 wherein the mapping step includes the following substeps:

(a) mapping relations, ~~such as tables~~, within the relational data model onto XML elements including a "table" and a "row" element;

(b) mapping tuples, ~~such as rows~~, within a relation onto XML elements which are nested within the XML "table" element; and

(c) mapping attributes ~~(i.e., columns)~~ of tuples onto XML elements which are nested within the XML "row" element.

Sub  
B1  
[ 26. Canceled

A1  
<sup>24</sup>  
27. (currently amended) The medium of claim <sup>22</sup>24 <sup>22</sup>[[26]] wherein, in order to perform a lossless mapping onto XML, further inserting ASN.1 tags into the XML specification by including one TAG element and three TAG attributes in the XML specification.

<sup>25</sup>  
28. (original) The medium of claim <sup>24</sup>27, wherein the XML TAG elements use an attribute to indicate their policy, their nature, and value of the tag and wherein, an additional attribute indicates whether IMPLICIT or EXPLICIT ASN.1 tagging has been used.

<sup>26</sup>  
29. (currently amended) The medium of claim <sup>25</sup>28 wherein additional attributes are included in the XML

A1  
specification, wherein XML tag attributes in a separate name-space Abstract Syntax Notation are ASN:policy ~~(EXPLICIT or IMPLICIT)~~, ASN:class ~~(UNIVERSAL, APPLICATION, CONTEXT, PRIVATE)~~, and ASN:tag containing the original ASN.1 tag value.

30. (new) The method of claim <sup>6</sup> 29, wherein in order to perform a lossless mapping onto XML, further inserting ASN.1 tags into the XML specification by including one TAG element and three TAG attributes in the XML specification.

31. (new) The method of claim <sup>6</sup> 30, wherein the XML TAG elements use an attribute to indicate their policy, their nature, and value of the tag and wherein, an additional attribute indicates whether IMPLICIT or EXPLICIT ASN.1 tagging has been used.

32. (new) The method of claim <sup>8</sup> 31, wherein additional attributes are included in the XML specification, wherein XML tag attributes in a separate name-space Abstract Syntax Notation are ASN:policy, ASN:class, and ASN:tag containing the original ASN.1 tag value.

33. (new) The medium of claim <sup>14</sup> 32, wherein in order to perform a lossless mapping onto XML, further inserting ASN.1 tags into the XML specification by including one TAG element and three TAG attributes in the XML specification.

34. (new) The medium of claim <sup>18</sup> 33, wherein the XML TAG elements use an attribute to indicate their policy, their nature, and value of the tag and wherein, an additional

attribute indicates whether IMPLICIT or EXPLICIT ASN.1 tagging has been used.

20  
35. (new) The medium of claim <sup>19</sup>~~34~~ wherein additional attributes are included in the XML specification, wherein XML tag attributes in a separate name-space Abstract Syntax Notation are ASN:policy, ASN:class, and ASN:tag containing the original ASN.1 tag value.